Staff Summary Sheet

	To	Action	Signature (Surname), Grade, Date		To	Action	Signature (Surname), Grade, Date
1	Col Butler	Approve	Butler O-6 9FEB15	6			
2	DFER	Review	Scott, AD25, 9 Feb 15	7			
3	Dr York	Action		8			
4				9			
5				10			

Grade and Surname of Action Officer	Symbol	Phone	Suspense Date
Dr York	DFEC	333-9193	13 Feb 2014

Subject		SSS Date
Clearance of Material for Public Release	Case Number:	6 Feb 2014

Summary

1. Purpose: To provide security and policy review on the paper at Tab 1 prior to release to the public.

2. Background:

- *Author(s)*: Scott Gruber, Robert Streeter, George York. (DFEC, x9193)
- *Title*: "Flexible Multi-agent Algorithm for Distributed Decision Making"

- *Release information*: To be presented at the *2015 INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS '15)*, June 9-12, in Denver, CO.

- *Previous clearance information*: None.

- *Recommended distribution statement*: Distribution A, Approved for public release, distribution unlimited.

3. Recommendation: Sign coord block above indicating document is suitable for public release. Suitability is based solely on the document being unclassified, not jeopardizing DoD interests, and accurately portraying official policy.

George York

Dr George York
Department of Electrical and Computer Engineering

Tab
1. Copy of Paper

Flexible Multi-agent Algorithm for Distributed Decision Making

Scott Gruber, Robert Streeter, George York*
Academy Center for Unmanned Aircraft Systems Research,
2354 Fairchild Drive, Suite 2F6, USAF Academy, CO 80840-6236
*Author to whom correspondence should be addressed; E-Mail: George.york@usafa.edu

Abstract—**This article presents an algorithm for decentralized control and decision making among a number of unmanned aircraft systems (UASs). The approach is scalable and adaptable to a variety of specific mission tasks. Additionally, the algorithm could easily be adapted for use on land or sea-based systems. Each UAS involved in an automated decision making process uses a selection of both predefined and observed parameters with corresponding predetermined weights to develop a score. These scores, and associated "scorecards," are then iteratively distributed to all involved agents, ranked, and used to determine task participants. Simulation as well as flight testing shows the algorithm provides correct team assignments in a variety of situations while remaining robust to agent dropouts, inconsistent communications, and dynamic situations.**

Index-words—**Unmanned aircraft systems, distributed, decentralized, decision making**

I. INTRODUCTION

Small, unmanned aerial aircraft systems (UASs) have been rising in popularity for a number of years. These platforms are more accessible since they are less expensive and easier to operate and maintain than their larger counterparts. Additionally, effectively using multiple small UASs requires cooperation between multiple active aircraft. This article introduces an algorithm, developed at the U.S. Air Force Academy's Center for Unmanned Aircraft Systems Research (ACUASR), which achieves automated, distributed decision making that correctly assigns tasks based on individual agent capabilities. The algorithm can be specifically tailored to meet mission objectives involving multiple agents with distinctly different capabilities.

Many organizations are realizing more and more uses for UASs. These applications will span a wide variety of disciplines outside the military arena [1-4]. Regardless of the application however, the ability to efficiently use multiple UASs will be critical [3, 5-8]. Early algorithm development simulations clearly indicate that more UAS agents in the field yields improved objective completion times. Figure 1 offers one example of simulated objective completion times for different numbers of deployed systems. Each line represents how a different number of active UASs (two, four, eight, or sixteen) deal with eight objectives. For this specific case, the mission involved locating and using two individual agents to track a specific object (the "objective") for a period of time before both agents return to a global search behavior. At least thirty-six simulation results were considered for each data point.

The maximum benefit of multiple UASs is achieved when agents can cooperate in the completion of mission objectives. To facilitate this cooperation, researchers at ACUASR have developed a decentralized decision making algorithm that takes mission specific parameters into account and selects the proper agent(s) for mission completion in dynamic scenarios. Numerous algorithms exist in the industry to handle autonomous decision making [9-11], but many have significant shortcomings such as relying too heavily on single points of failure [12] (an increasing number of developing algorithms have adopted decentralized approaches, thus overcoming this weakness), failure to accommodate multiple participating agents in the task assignment [13], or depending

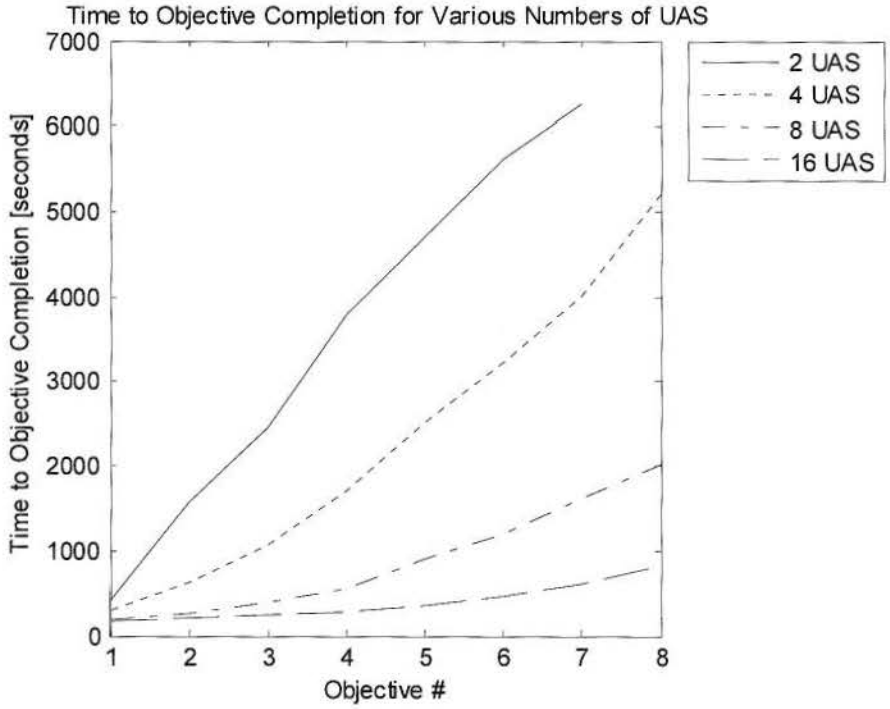

Fig. 1. Objective completion times for varying number of active agents in simulation.

upon reliable, predictable communications. An assessment of the state of the industry appears in [14]. The algorithm presented herein overcomes these obstacles, allowing for flexible, decentralized decisions to be made by a cooperating group of heterogeneous agents in an uncertain communications environment.

II. METHODS AND MATERIALS

Allowing UASs to execute complex missions with minimal human intervention requires automated decision making on-board each craft. The algorithm presented here uses scoring of available resources, which enables each involved UAS to independently analyze a situation and develop task assignments. Ground-station operators can review the proposed assignments prior to task execution. The process is iterated a preset number of times to help achieve consensus and allow all UASs the opportunity to consider all available resources before assigning task roles.

This section will provide an algorithm overview, offer underlying mathematical details, and provide testing approaches used during development. The results observed during both hardware-in-the-loop (HIL) and flight tests will be discussed in a later section.

A. Algorithm Overview

The decision making process involves several distinct stages (as illustrated in Figure 2), and begins with each UAS subjectively analyzing the situation at hand. There are two ways in which a UAS might become involved with the decision process. First, if a UAS triggers an objective (e.g., locates a potential target) it will consider whether it is available and appropriate to begin working on the task associated with that objective. This is referred to as internal *self-assessment*. Second, if a UAS receives a request for assistance from another vehicle, it will also perform an initial *self-assessment*, in this case caused by an outside source.

Self-assessment considers the current state of the UAS and determines whether participation in the task being evaluated is viable. If a discovering agent is available and appropriate to begin handling a newly activated objective, it proceeds on to the *populate scorecard* state, otherwise the location of the objective is shared with other agents and the discovering UAS proceeds with its current tasking.

Populating a scorecard involves considering locally available information only. A score is computed based on selected parameters. These parameters can be dynamic, and therefore must be observed (such as time to intercept, remaining

flight time, etc.), or static and thus predetermined (such as agent priority, objective priority, etc.). Once the discoverer has computed an initial score, it proceeds to a one-time self-check called *single-UAS assignment*. The discoverer will only conduct this test once, and any non-discovering (secondary) UAS will not enter this phase at all.

In *single-UAS assignment*, the UAS checks to see if it can perform all the necessary duties associated with the objective by itself. If so, no request for participation from other agents is issued, and the discoverer immediately notifies the mission control station (MCS) operator of the intent to begin handling the objective[1]. If the discoverer determines it cannot complete the task by itself, it sends a participation request and its scorecard to all other agents. This initiates the *self-assessment* process in those agents. The discovering agent then proceeds to *group-assessment*. Recipients of the participation request enter the *self-assessment* phase.

During the *group-assessment* phase of the algorithm, information received from each individual agent is considered. As more agents conduct *self-assessments* and choose to participate, more scorecards become available for consideration. These scorecards are rank ordered based on the score computed during the *populate scorecard* state. Then, beginning from the top, each scorecard is considered until sufficient resources for completing the objective are identified. The UAS with the best score is considered first, the second-best is next, and so on. In this way, those agents with the best scores receive the task assignments first. The process is considered complete once the UAS resources thusly assigned are adequate for the task. To provide redundancy for communication losses, promote convergence, and ensure adequate situational awareness, the processes of *self-assessment*, *populate scorecard*, and then *group-assessment* are repeated a preset number of times or until a timer is exhausted. During this looping, each vehicle shares its scorecard multiple times with all other vehicles involved in the decision. The MCS is also included as a recipient, so that the decision process can be monitored by the human operator. Once the process ends, each agent has a complete collection of up-to-date scorecards, (collectively referred to as a scoresheet) enabling a decision to be made about *multi-UAS assignment*.

Once the *group-assessment* process is complete, the agents transition to the *multi-UAS assignment* portion of the algorithm. Agents not involved with the task continue with their previous behavior or return to a predetermined default behavior (such as a wide area search). All UAS provisionally assigned to the task send a confirmation request to the MCS operator for action. The MCS operator can approve the request as developed by the UASs, alter the role assignments, or reject the tasking altogether[2]. If the operator approves the role assignment, participating agents begin accomplishing the task objective. However, if the operator rejects the tasking, all UASs that participated in the decision will continue without changing behavior. If scorecards are mismatched between agents and the MCS, scorecard resend requests can be sent[3]. This ensures all parties have identical information.

Figure 2 provides a flow diagram of the algorithm. The green object in the upper left indicates the onset or entry point of the algorithm, and the orange objects are the possible outcomes. Each subsection of the algorithm (*self-assessment*, *populate scorecard*, *group-assessment*, *single-UAS assignment*, and *multi-UAS assignment*) is delineated for clarity.

[1] This step could, if required, involve active operator approval. Whether or not a specific application requires operator confirmation should be decided on a case-by-case basis. It should also be noted that the MCS is different than a ground control station (GCS). Where the GCS handles flying and flight monitoring, the MCS supervises mission specific tasking and approves task assignments.

[2] The ability to edit the role assignments has not yet been implemented or tested, but is designed to simply transmit new assignments from the MCS to the involved agents.

[3] The resend request relies on matching of computed checksums onboard each UAS and at the MCS. This feature has not yet been implemented. In mature laboratory and flight tests, checksums between involved agents matched.

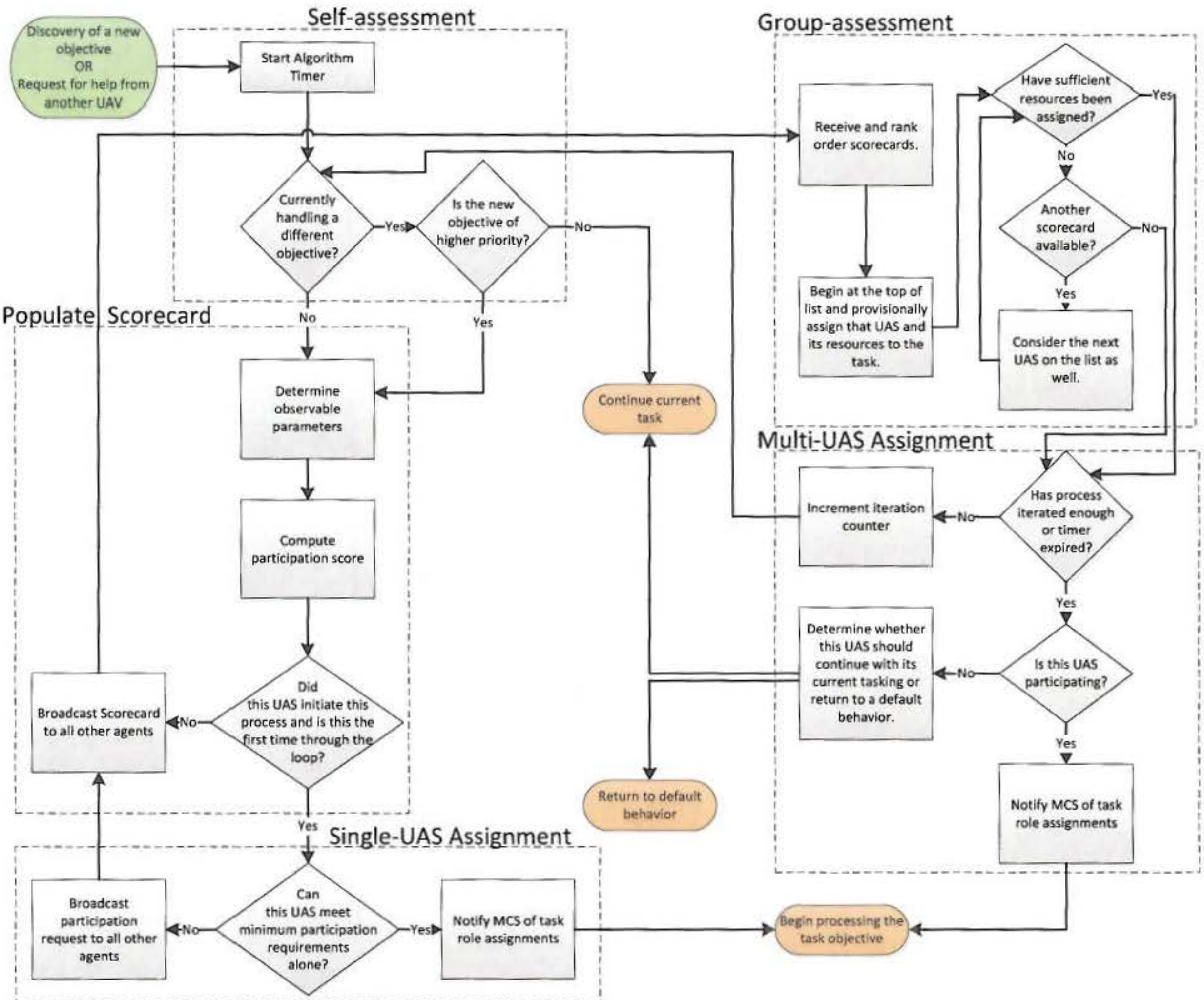

Fig. 2. Decision Algorithm Flow Diagram

B. Algorithm Mathematics

Mathematically, the algorithm is very basic and begins by defining a few fundamental values. First, specific mission parameters $\boldsymbol{P}$ and an equal-length set of associated decision weights $\boldsymbol{W}$, as shown in Equation (1), are selected.

$$\begin{aligned} \boldsymbol{P} &= \{p_1, p_2, p_3, \ldots, p_n\} \\ \boldsymbol{W} &= \{w_1, w_2, w_3, \ldots, w_n\} \end{aligned} \tag{1}$$

Values within $\boldsymbol{P}$ can be predetermined (such as vehicle priority, objective priority, etc.) or they may be observable in real-time (such as time to arrive, remaining flight time, etc.). The decision weights, $\boldsymbol{W}$, are set during system initialization. Equation (2) describes how $\boldsymbol{P}$ and $\boldsymbol{W}$ are combined in parallel to compute a score, S. This score is used to rank order each UAS: a lower score indicates a less desirable participant.

$$S = \sum_{k=1}^{n}(w_k * p_k) \tag{2}$$

Given a rank ordered list of available UAS, individual vehicle capabilities are considered to ensure that adequate resources are assigned to the objective. These capabilities are mission specific and could involve available sensor type(s) and flight capabilities such as vertical take-off and landing (VTOL), deployable persistent sensor capability, etc. Table 1 offers an example arrangement of UAS parameters.

TABLE 1. EXAMPLE UAS OBJECTIVE PARAMETER ARRANGEMENT

Capability	Representing Parameters	
	Objective Type 1	**Objective Type 2**
Sensor Type	s_1	s_2
VTOL Status	v_1	v_2
Deployable Sensor Status	d_1	d_2

The values for each UAS's parameters might vary depending upon the objective, thus, each parameter is defined separately for each objective type. For instance, a particular UAS might be equipped with a foliage penetrating sensor and another UAS with a high speed tracking sensor. If a Type 1 objective is to locate a lost person possibly under tree cover, the UAS with foliage penetrating capability would be more desirable than a UAS with high speed tracking. To account for this, each UAS has its own set of parameters for each objective type.

In addition to capability parameters, a minimum probability of success, $P_{s,min}$ is also assigned for each objective. This defines, in general terms, the resources required to complete an objective. While the capabilities of each UAS might differ, $P_{s,min}$ is the same across all platforms for a specific objective. As resources are provisionally allocated to an objective (part of the *group-assessment* process), the accumulated probability of success, $P_{s,acc}$, increases from zero. Once $P_{s,acc}$ meets or exceeds $P_{s,min}$, the task role assignment is considered complete. This may only require the participation of a single UAS or it may require the collaboration of a number of aircraft. The algorithm is flexible and allows allocation of all needed up to all available resources. Values of $P_{s,acc}$ for specific resource configurations are stored in a look-up table. Each entry in the table corresponds to a different configuration, as they relate to the objective type being considered. If these combined probabilities can be mathematically derived prior to launching a mission, these resource configurations can also be derived as needed

Having sufficient resources to completely handle a given objective could be an insurmountable logistics issue. To best accommodate this situation, the algorithm will assign all available resources if necessary. This ensures that, while $P_{s,acc}$ may be less than $P_{s,min}$, a "best effort" will always be available. Regardless of whether or not sufficient resources are successfully identified, once enough iterations have completed or the consensus timer expires, approval is requested from the MCS operator and the decision process is considered complete. This request for approval provides human-in-the-loop capability, and may not be required. It is included to accommodate the evolving autonomy of UASs.

III. RESULTS AND DISCUSSION

Algorithm design, validation, and testing took place in three primary ways. Initially, MATLAB simulations were created implementing an early version of the algorithm. This provided initial values for timeouts, counters, and weights based on actual results specific to constructed scenarios. Throughout development, hardware-in-the-loop (HIL) testing was conducted. This allowed actual UAS hardware (including cameras, autopilots, and on-board computers) to be utilized during testing, without the logistics and other inherent stresses associated with a flight test. However, laboratory testing alone cannot fully demonstrate the robust functionality of the algorithm. Therefore, two live flight tests were conducted.

This section discusses the MATLAB simulation design work, the HIL testing, and the flight testing. After completion of the MATLAB simulations, the algorithm was implemented using C# to run on a single-board computer (SBC) onboard each aircraft. Additionally, proprietary MCS software was developed to interface with the airborne SBCs. This program, called the Core UASs Control Station (CUCS) provides an intuitive, flexible, and real time mission control interface capable of monitoring up to four UASs. Figure 3 is a mock-up screen capture of what CUCS would look like during a live, 4-UASs mission.

A. MATLAB Simulation Design

Preliminary design testing in MATLAB enabled flight-ready implementation of the algorithm to be more robust and flexible. A wide variety of scenarios were tested, ranging from two UAS agents and one objective, up to sixteen agents and sixteen objectives. Using the data

from hundreds of simulations, several initial values were selected for the C# implementation of the algorithm. These included decision weights, decision iteration limits, and time-out values. Figure 1 was derived from data gathered during these simulations.

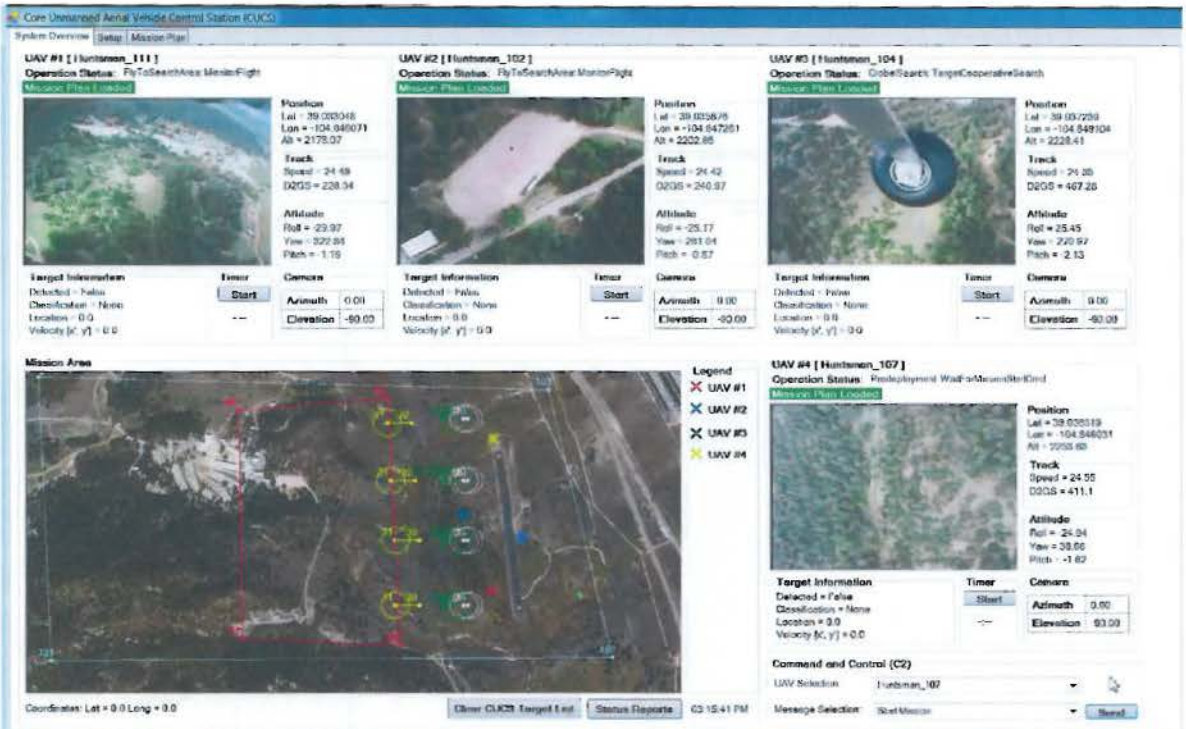

Fig. 3. Core UAS Control Station screen mock-up

B. Hardware-in-the-Loop Testing

Hardware-in-the-loop simulation allows testing in a highly controlled laboratory setting using the actual aircraft payload, including flight-grade avionics and simulated inertial measurement unit (IMU) sensor data. Situation parameters, such as number of active aircraft and the configuration of each craft, can be easily changed to test different possible arrangements. This process was extensively used during the development of the algorithm and provided invaluable debugging information. Additionally, analyses of the communication traffic and interactions with the MCS helped ensure no in-flight system issues.

The most recent iteration of HIL testing showed the algorithm functioning as intended with four UASs in three different capability configurations. The algorithm correctly arrived at the optimal solution given the situation, and task objective completion was achieved. In the case of this specific test, two separate UASs were needed for the task, each with different payload capabilities. These aircraft were selected by the algorithm based on their scores at the time the decision was made and because their combined resources were sufficient for task completion. The other two vehicles participated in the decision process, and, when not assigned active roles, correctly continued their previous behavior.

A number of other scenarios were tested throughout algorithm development. These included intentionally shorthanded UAS deployments, a variety of matching and differing UAS configurations, and stationary and mobile objectives. In this way, a variety of situations could be accounted for and algorithm response correctly adjusted to ensure robustness regardless of situation.

C. Flight Testing

Two separate flight tests were conducted to demonstrate, in a live environment, algorithm functionality. The first test was done with three UASs early in algorithm implementation and resulted in only partial success. Handling of multiple tasks had not been adequately considered when the algorithm was implemented. This resulted in one UAS being

assigned to multiple objectives. This was not handled properly and caused one objective to be lost. Additionally, the implementation for requesting approval from the MCS was not robust and allowed participating systems to develop and get approved differing task role assignments. While the algorithm did not need to change, the implementation demanded attention and work was done to remedy it.

After improving the algorithm implementation and further HIL testing, the second flight test was conducted. This test used only two UASs, but successfully demonstrated the algorithm's capabilities in flight. An objective was identified, the decision process correctly triggered, and both systems arrived at the same optimal solution.

There are issues that can only be properly accounted for while in-flight. These include dynamic environmental situations and communication realities. By performing not only laboratory testing, but also flight testing, the validity of the algorithm is more solidly affirmed.

D. Notable Areas for Improvement

While the algorithm functions correctly as-implemented, the literature suggests a potential point of improvement. ACUASR's algorithm relies on repeated messaging concerning situational awareness and decision participation in order to function. Using a more asynchronous approach, as the authors in [15] suggest, could further improve the adaptability and robustness of our algorithm.

Additionally, researchers at the ACUSAR propose a method to accelerate the task role assignment process: use scorecard checksum comparisons to identify an agreed upon solution prior to the algorithm iteration count or the completion timer indicating an end to the process. Currently, the process continues until the iteration limit is reached, even if convergent scorecards are computed prior to that. This improvement would require storing the scoresheet associated with a particular checksum so that it could be recalled later. If, at some point during the process, a minimal required number of involved agents arrive at the same workable solution (as determined by matching checksums), that solution could be accepted and the process truncated. This could lead to a sub-optimal solution, but task role assignment would be faster. By reviewing HIL simulation logs, this change could improve decision time by almost 40% for two UAS operations and about 23% for three UAS operations. With the iteration limit set at three, no significant improvement could be realized for four UAS operations. This improvement has not yet been implemented, but shows promise as a robust method for speeding up the decision making process.

IV. CONCLUSIONS

Researchers at the U.S. Air Force Academy's Center for Unmanned Aircraft Systems Research successfully developed, implemented, and tested a distributed and decentralized decision making algorithm. The algorithm remains robust under the rigors of multiple-unmanned aerial vehicle coordination while staying flexible with regard to task and agent participation requirements. Hardware-in-the-loop and flight testing proved that the algorithm properly provides task assignments while operating on actual flight hardware.

ACKNOWLEDGMENT

The authors would like to heartily acknowledge the contributions of other members (current or former) of the Academy Center for Unmanned Aircraft Systems Research: Dr. Erlind Royer (BGen, USAF, ret), Dr. Daniel Pack (former director of the ACUASR, now Electrical and Computer Engineering Department Chair at the University of Texas in San Antonio), Dr. Stan Baek (former researcher at the ACUASR, now Assistant Professor of Electrical Engineering at the University of Michigan in Dearborn), Dr. Harry Direen (researcher at the ACUASR), Kelly McMurtrey (researcher at the ACUSAR), and Dr. Ricky Sward (system engineer at MITRE Corporation and our test pilot).

REFERENCES

[1] Abrahamsen, H.B. How to improve situation assessment and decision-making in a simulated mass casualty incident by using an unmanned aerial vehicle. *Scandinavian Journal of Trauma, Resuscitation and*

Emergency Medicine. 2014, vol. *22*, Supplement 1/pp 2.

[2] Singer, P. Unmanned Aircraft Systems: Perceptions & Potential. Aerospace Industries Association technical report. 2013.

[3] Phan, C.; Liu, H.H.T. A Cooperative UAV/UGV Platform for Wildfire Detection and Fighting. Proceedings of the International Conference on System Simulation and Scientific Computing, Beijing, China, October 10-12, 2008.

[4] Parnell, B.A. Five Nonlethal Uses for Drones Aside From Amazon Deliveries. Forbes Magazine. Available online: http://www.forbes.com/sites/bridaineparnell/2013/12/02/five-nonlethal-uses-for-drones-aside-from-amazon-deliveries/. 2013.

[5] Khan, A.; Yanmaz, E.; Rinner, B. Information Merging in Multi-UAV Cooperative Search. Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, May 31, 2014 to June 7, 2014.

[6] Cummings, M.L.; Clare, A.; Hart, C. The Role of Human-Automation Consensus in Multiple Unmanned Vehicle Scheduling. *Human Factors: The Journal of the Human Factors and Ergonomics Society*. 2010, vol. 52-1, pp. 17-27.

[7] Meng, W.; He, Z.; Teo, R.; Su, R.; Shehabinia, A. R.; Lin, L.; Xie, L. Decentralized Control of Multi-UAVs for Target Search, Tasking, and Tracking. Proceedings of the World Congress of The Internationsl Federation of Automatic Contol, Cape Town, South Africa, August 24-29, 2014.

[8] Ahmed, A.; Patel, A.; Brown, T.; Ham, M.J.; Jang, M.W.; Agha, G. Task Assignment for a Physical Agent Team via a Dynamic Forward/Reverse Auction Mechanism. University of Illinois Technical Report, UIUCDCS-R-2004-2507, December 2004.

[9] Gan, S.K.; Sukkarieh, S. Multi-UAV Target Search Using Explicit Decentralized Gradient-based Negotiation. Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, May 9-13, 2011.

[10] Oh, G.; Kim, Y. Market-based Decentralized Task Assignment for Cooperative UAV Mission Including Rendezvous. Proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, August 19-22, 2013.

[11] Eun, Y.; Bang, H. Cooperative Task Assignment/Path Planning of Multiple Unmanned Aerial Vehicles Using Genetic Algorithms. *Journal of Aircraft*. 2009, vol. 46-1, pp. 338-343.

[12] M.B. Dias and A. Stentz. A Market Approach to Multirobot Coordination. Carnegie Mellon Robotics Institute Technical Report, CMU-RI-TR-01-26, August 2001.

[13] Brunet, L.; Choi, H. L.; How, J. P. Consensus-Based Auction Approaches for Decentralized task Assignment. Proceedings of the AIAA Guidance, Navigation, and Control Conference, Honolulu, Hawaii, August 18-21, 2008.

[14] Hagelin, P. Analysis of Centralized and Decentralized Communication and Decision Making for Cooperating Autonomous UAVs. Master's thesis, University of Linkpöping, 2011.

[15] Johnson, L.B.; Ponda, S.S.; Choi, H.L.; How, J.P. Improving the Efficiency of a Decentralized Tasking Algorithm for UAV Teams with Asynchronous Communications. Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, Canada, August 2-5, 2010.